

REMARKS

Claims 1-26, 38-49, and 55-64 are pending in the application. Claims 1-26, 38-49, and 55-64 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Application Publication No. 2003/0163479 to Matthews et al. in view of U.S. Patent Application Publication No. 2003/0140027 to Huttel et al.

Reconsideration is requested. The rejections are traversed. No new matter is added. Claims 1, 7, 10-11, and 17 are amended. Claims 1-26, 38-49, and 55-64 remain in the case for consideration.

INTERVIEW SUMMARY

On July 19, 2007, the undersigned held a telephone interview with Examiner Robinson. The claims and references were discussed. No agreement was reached.

The undersigned argued that the OASM specification of Matthews would not be able to define the data store as recited for example in claim 38. For example, the OASM specification does not support the concept of a generic document, as recited in the claims. The undersigned also argued that Matthews did not teach or suggest the concept of multiple schemas, as recited for example in claim 17.

The undersigned further argued that the recitation of multiple data stores, for example in claim 9, was also not taught by Matthews: while Matthews mentioned the possibility of multiple data stores, Matthews did not teach or suggest the use of a key to reference another data store.

The Examiner indicated that she would consider the arguments made. The Examiner suggested that the claims would be better distinguished from the reference by including discussion about the use of the data stores. The undersigned pointed out that such claims had been originally present in the application as filed, but were restricted out by the Examiner in the Restriction Requirement dated September 18, 2006.

REJECTIONS UNDER 35 U.S.C. § 103(a)

Matthews and Huttel generally

Matthews teaches a method and apparatus for implementing a data management system. An object relationship model is created automatically using a specification of the data storage schema. The object relationship model defines views that limit how consumers can interact with the data store

and validate that the data store is being used properly. Consumers interact with the data store through the object relationship model through the views.

Huttel teaches a universal programming interface. Huttel uses tables that have only one key to distinguish a record. Huttel permits database operations that can be either simple or complex.

The Examiner cites to Huttel only for the concepts of a key value field assigned to a particular record (*see*, Office Action dated May 9, 2007, page 3), keys (*see*, Office Action dated May 9, 2007, pages 4 and 6), an XML schema (*see*, Office Action dated May 9, 2007, pages 5 and 6). The Applicant believes that Huttel teaches no other features of the claims.

Matthews and Huttel do not teach a schema as claimed or a database defined from a schema for a generic document

Claim 1 is directed toward a computer, comprising: a memory; and a data structure to store a schema for a generic document, the data structure stored in the memory and including: a definition of a first element, the definition of the first element including an element value field; and a key identifier to identify a key value field to be used as a key in a data store.

Claim 10 is directed toward a computer according to claim 1, wherein the data structure is operative to store an eXtensible Markup Language (XML) schema.

Claim 11 is directed toward a computer system, comprising: a data store to store a first generic document and to store a value for a first key value field, the value loaded from the first generic document; and a first schema applicable to the first generic document, the first schema including: a definition of a first element, the definition of the first element including an element value field; a first key identifier to identify the first key value field in the first generic document to be used as a key in a data store.

Claim 26 is directed toward a system according to claim 11, wherein: the first generic document is an eXtensible Markup Language (XML) document; and the first schema is an XML schema.

Claim 38 is directed toward a method for defining a data store in a computer, comprising: accessing a schema; locating an object defined in the schema as a key; defining a first data structure in the data store for the object; identifying the first data structure in the data store as a key data structure; and defining a second data structure in the data store for a generic document conforming to the schema.

Claim 42 is directed toward a method according to claim 38, wherein: the generic document is an eXtensible Markup Language (XML) document; and the schema is an XML schema.

Claim 43 is directed toward a method for defining a schema for a generic document in a computer, comprising: defining a first element in the schema, the first element including an element value field; and identifying a key value field in the schema to be used as a key in a data store.

Claim 49 is directed toward a method according to claim 43, wherein the schema is an XML schema.

Claim 55 is directed toward a computer-readable medium containing a program to define a data store, comprising: software to access a schema; software to locate an object defined in the schema as a key; software to define a first data structure in the data store for the object; software to identify the first data structure in the data store as a key data structure; and software to define a second data structure in the data store for a generic document conforming to the schema.

Claim 57 is directed toward a computer-readable medium according to claim 55, wherein: the generic document is an eXtensible Markup Language (XML) document; and the schema is an XML schema.

Claim 58 is directed toward a computer-readable medium containing a program to define a schema for a generic document, comprising: software to define a first element in the schema, the first element including an element value field; and software to identify a key value field in the schema to be used as a key in a data store.

Claim 64 is directed toward a computer-readable medium according to claim 58, wherein the schema is an eXtensible Markup Language (XML) schema.

As discussed above in the Interview Summary, the focus of Matthews is not to define a data store to support the use of generic documents, but rather to explain how an Object Access and Storage Model (OASM) can be used to define an interface to a database that meets the definition of the OASM. According to Matthews, “[t]he present invention uses the OASM specification to generate an object model interface and data store schema, which are realized by the DAC and a data store at runtime” (see Matthews, ¶ 53). In other words, in Matthews, the starting point is the OASM specification; from this, the interface and data store schema are generated, and the DAC is used to implement the interface automatically.

Compare this description with the invention as claimed. Claims 38 and 55, for example, recite “accessing a schema” and “defining a second data structure in the data store for a generic document conforming to the schema”, all without reference to an OASM. Without the OASM in Matthews, nothing can be done: the OASM is a necessary starting point to the process of Matthews. But the OASM is not needed to enable the invention.

In fact, Matthews does not care about the definition of the database. Matthews, as noted above, is concerned with being able to define an interface to a database automatically. Such automatic construction of an interface is possible if the structure of the database is known; if the structure of the database is not known, no such interface construction is possible.

Claims 1, 43, and 58 relate to the definition of a schema for a generic document; claims 38 and 55 relate to the definition of a data store (that is, the structure of the data store) based on such a schema. If Matthews presupposes that the definition of the database structure (that is, the OASM model) is known in advance, Matthews must assume that constructs, such as those recited in claims 1, 55, and 58, already exist: Matthews does not explain how such constructs are built. Similarly, as Matthews assumes the database model is already defined, Matthews assumes that the methods recited in claim 38 and 55 have already been carried out. And all of this assumes that Matthews is building interfaces to databases that manipulate generic documents as claimed. In fact, Matthews does not describe how the OASM model is defined, because Matthews explains how the interface can be constructed for any OASM-compliant database. If Matthews had described his idea only in the context of data stores similar to those being defined and used in this patent application, Matthews would have significantly limited the application of his idea.

Further, Matthews does not teach a schema for a generic document at all. For example, claim 1 recites “a data structure to store a schema . . . including: a definition of a first element, the definition of the first element including an element value field; and a key identifier to identify a key value field to be used as a key in a data store”; claims 43 and 58 are similar. The Examiner cites to ¶¶ 53, 60, 67-68, and 75 of Matthews as teaching a first element including an element value field, and a key identifier, and to ¶ 1687 of Huttel as teaching a key value field. The Applicant points out that these features are all discussed in both Matthews and Huttel with reference to a database, and not as part of a schema for a document. As document schemas and

databases are completely different objects, it would not have been obvious to a person skilled in the art to take database concepts and apply them to document schemas.

The Applicant also notes that while there are common terms (e.g., “schema”) between the claimed invention and Matthews, these terms do not refer to the same objects. In Matthews, the “schema” is a “data store schema”: in other words, it defines the data store. In contrast, in the claimed invention, the schema determines a format for a document: it is the document that needs to comply with the schema, not the data store. This is recited in the claims in numerous places: for example, claim 1 (“schema for a generic document”); claim 11 (“first schema applicable to the first generic document”); claims 38 and 55 (“a generic document conforming to the schema”); and claims 43 and 58 (“a schema for a generic document”). Further, claims 10, 26, 42, 49, 57, and 64, which depend from independent claims 1, 11, 38, 43, 55, and 58, respectively, recite that the schema is an eXtensible Markup Language (XML) schema. XML schemas do not describe data stores: XML schemas describe document formats.

The Examiner rejects claims 10, 26, 42, and 49 by citing to Huttel for the concept of an XML schema. The Applicant points out that Huttel is using the term “schema” in reference to a document, which is different from Matthews’s use of the term “schema” in defining a database. Because of the inconsistent uses of the term “schema” in Matthews and Huttel, it would not be obvious to combine Matthews and Huttel as the Examiner suggests: these “schemas” are not interchangeable, nor can the schema of Huttel be used in defining a database in Matthews.

The Examiner has rejected claim 55-64 on the same rationale as claim 1-26 and 38-49, without any specificity. Accordingly, all arguments presented with reference to claims 1, 10-11, 26, 38, 42-43, and 49 also apply to claims 55, 57-58, and 64 for the same reasons.

As the combination of Matthews and Huttel does not teach a schema for a generic document that identifies key value fields, how to define such a schema, a database defined based on such a schema, how to define such a database, or that such a schema can be an XML schema, claims 1, 10-11, 26, 38, 42-43, 49, 55, 57-58, and 64 are patentable under 35 U.S.C. § 103(a) over Matthews in view of Huttel. Accordingly, claims 1, 10-11, 26, 38, 42-43, 49, 55, 57-58, and 64 are allowable, as are dependent claims 2-9, 12-25, 39-41, 44-48, 56, and 59-63.

Matthews and Huttel do not teach a schema that identifies keys as properties or attributes

Claim 2 is directed toward a computer according to claim 1, wherein: the definition of the first element includes the key identifier as a property of the first element; and the key identifier identifies the element value field as the key value field.

Claim 3 is directed toward a computer according to claim 1, wherein: the definition of the first element includes a definition of a first attribute of the first element, the definition of the first attribute including the key identifier; and the key identifier identifies the element value field as the key value field.

Claim 5 is directed toward a computer according to claim 4, wherein: the definition of the second attribute includes the key identifier as a property of the second attribute; and the key identifier identifies the attribute value field as the key value field.

Claim 6 is directed toward a computer according to claim 4, wherein: the definition of the first element further includes a definition of a third attribute of the first element, the definition of the third attribute including the key identifier; and the key identifier identifies the attribute value field of the second attribute as the key value field.

Claim 12 is directed toward a system according to claim 11, wherein: the definition of the first element includes the first key identifier as a property of the first element; and the first key identifier identifies an element value field of the first element as the first key value field.

Claim 13 is directed toward a system according to claim 11, wherein: the definition of the first element includes a definition of a first attribute, the definition of the first attribute including the first key identifier; and the first key identifier identifies an element value field of the first element as the first key value field.

Claim 15 is directed toward a system according to claim 14, wherein: the definition of the second attribute includes the first key identifier as a property of the second attribute; and the first key identifier identifies the attribute value field as the first key value field.

Claim 16 is directed toward a system according to claim 14, wherein: the definition of the first element further includes a definition of a third attribute, the definition of the third attribute including the first key identifier; and the first key identifier identifies the attribute value field as the first key value field.

Claim 44 is directed toward a method according to claim 43, wherein: defining a first element includes assigning a property to the first element as the key; and identifying a key value field includes identifying the element value field as the key value field.

Claim 45 is directed toward a method according to claim 43, wherein: defining a first element includes defining a first attribute of the first element, the first attribute identifying the first element as the key; and identifying a key value field includes identifying the element value field as the key value field.

Claim 47 is directed toward a method according to claim 46, wherein: defining a second attribute includes assigning a property to the second attribute as the key; and identifying a key value field includes identifying the attribute value field as the key value field.

Claim 48 is directed toward a method according to claim 46, wherein: defining a second attribute includes defining a third attribute of the first element, the third attribute identifying the second attribute as the key; and identifying a key value field includes identifying the attribute value field of the second attribute as the key value field.

Claim 59 is directed toward a computer-readable medium according to claim 58, wherein: the software to define a first element includes software to assign a property to the first element as the key; and the software to identify a key value field includes software to identify the element value field as the key value field.

Claim 60 is directed toward a computer-readable medium according to claim 58, wherein: the software to define a first element includes software to define a first attribute of the first element, the first attribute identifying the first element as the key; and the software to identify a key value field includes software to identify the element value field as the key value field.

Claim 62 is directed toward a computer-readable medium according to claim 61, wherein: the software to define a second attribute includes software to assign a property to the second attribute as the key; and the software to identify a key value field includes software to identify the attribute value field as the key value field.

Claim 63 is directed toward a computer-readable medium according to claim 61, wherein: the software to define a second attribute includes software to define a third attribute of the first element, the third attribute identifying the second attribute as the key; and the software

to identify a key value field includes software to identify the attribute value field of the second attribute as the key value field.

In rejecting claims 2-3, 5-6, 12-13, 15-16, 44-45, 47-48, 59-60, and 62-63, the Examiner cites to the OASM model of Matthews. As argued above, the OASM model of Matthews is used to define a database. In contrast, claims 2-3, 5-6, 12-13, 15-16, 44-45, 47-48, 59-60, and 62-63 relate to schemas for generic documents, and where identifiers of keys can be found. As the OASM model of Matthews applies only to databases, it cannot be used with document schemas. Accordingly, claims 2-3, 5-6, 12-13, 15-16, 44-45, 47-48, 59-60, and 62-63 are patentable under 35 U.S.C. § 103(a) over Matthews in view of Huttel, and are therefore allowable.

Matthews and Huttel do not teach keys referencing a second data store

Claim 9 is directed toward a computer according to claim 8, wherein: the key identifier identifies the key value field as a foreign key for the data store; and the key identifier references a second data store.

As discussed above, Matthews is concerned with interfaces to a database, and Huttel is concerned with a universal programming interface. Neither Matthews nor Huttel teaches or suggests keys referencing other data stores. Matthews teaches that the DAC can be used to interface with multiple databases, but does not disclose that the databases themselves refer to each other. Huttel discloses foreign keys in the context of referencing from one table to another (*see* Huttel, ¶ 12 (“A foreign key in a table references a primary key in another table”)), but not across databases. As the term “data store” is concerned equivalent to “database” (*see* specification, page 6, lines 5-6), a table in Huttel is not equivalent to a data store as recited in the claims.

As the combination of Matthews and Huttel does not teach or suggest multiple data stores, claim 9 is patentable under 35 U.S.C. § 103(a) over Matthews in view of Huttel. Accordingly, claim 9 is allowable.

Matthews and Huttel do not teach a data store that can store generic documents defined by different schema

Claim 17 is directed toward a system according to claim 11, wherein: the data store is operative to store a second generic document and to store a second value for a second key value

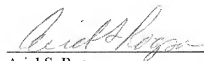
field, the value loaded from the first generic document; and the system further comprises a second schema applicable to the second generic document, the second schema including: a definition of a second element, the definition of the second element including an element value field; a second key identifier to identify the second key value field in the second generic document to be used as a key in a data store.

As discussed above, the database in Matthews is not defined based on schemas for documents. Accordingly, if Matthews does not teach a data store capable of storing a generic document defined by one schema, it cannot teach a data store capable of storing generic documents defined by more than one schema. Huttel does not correct this deficiency. Therefore, claim 17 is patentable under 35 U.S.C. § 103(a) over Matthews in view of Huttel. Accordingly, claim 17 is allowable, as is dependent claim 18.

For the foregoing reasons, reconsideration and allowance of claims 1-26, 38-49, and 55-64 of the application as amended is requested. The Examiner is encouraged to telephone the undersigned at (503) 222-3613 if it appears that an interview would be helpful in advancing the case.

Respectfully submitted,

MARGER JOHNSON & McCOLLOM, P.C.



Ariel S. Rogson

Reg. No. 43,054

MARGER JOHNSON & McCOLLOM, P.C.
210 SW Morrison Street, Suite 400
Portland, OR 97204
503-222-3613
Customer No. 45842